

AMENDMENTS TO THE CLAIMS

Upon entry of this amendment, the following listing of claims will replace all prior versions and listings of claims in the pending application.

Listing of Claims

Please amend claims 1, 6, 8, 11, 13, 15 and 16 as follows:

1. (Currently Amended) A computer-implemented method for efficiently parsing received data files, comprising:

receiving, by a virtual browser executing on a device intermediary to a plurality of clients and a plurality of web servers, a data file from a web server responsive to a request by a client of the plurality of clients;

retrieving, by the virtual browser, a stored version of the data file and a syntax tree comprising nodes and tokens representing data within the data file, the tree including at least one static node;

comparing, by a comparison engine in communication with the virtual browser, the stored version of the data file with the received data file to identify non-matching content in the received data file;

parsing, by a parsing engine of the virtual browser, only the non-matching content of the received data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file;

replacing, by the virtual browser, at least one static node of the syntax tree with a token;
and

creating, by the virtual browser, a mapping from each token to the at least one of the subtrees in the syntax tree; and

transmitting, by the device, the syntax tree to the client responsive to the request.

2. (Canceled)

3. (Canceled)

4. (Previously Presented) The computer-implemented method of claim 1 wherein the data file is a web page.

5. (Previously Presented) The computer-implemented method of claim 1 wherein the data file is an HTML file.

6. (Currently Amended) A method for efficiently parsing web pages, comprising:

receiving, by a virtual browser executing on a device intermediary to a plurality of a clients and a plurality of web servers, a first HTML page from a web server responsive to a request by a client of the plurality of clients;

retrieving, by the virtual browser, a cached version of the HTML page and a syntax tree comprising nodes and tokens representing data within the first HTML page, the tree including at least one static node;

comparing, by a comparison engine in communication with the virtual browser, the cached version of the HTML page with the received HTML page to identify non-matching content in the received HTML page;

parsing, by a parsing engine of the virtual browser, only the non-matching content in the received HTML page to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file;

replacing, by the virtual browser, at least one static node of the syntax tree with a token;
and

creating, by the virtual browser, a mapping from each token to the at least one of the subtrees in the syntax tree; and

transmitting, by the device the syntax tree to the client responsive to the request.

7. (Canceled)

8. (Currently Amended) A method for efficiently parsing HTML pages, comprising:

receiving, by a virtual browser executing on a device intermediary to a plurality of a clients and a plurality of web servers, a first HTML page from a web server responsive to a request by a client of the plurality of clients;

responsive to a determination that a cached version of the HTML page exists:

retrieving, by the virtual browser from a cache, the cached version of the HTML page and a first syntax tree comprising nodes and tokens representing data within the first HTML page, the first tree including at least one static node;

comparing, by a comparison engine in communication with the virtual browser, the cached version of the first HTML page with the received HTML page to identify non-matching content in the received HTML page;

parsing, by a parsing engine of the virtual browser, only the non-matching content to form a subtree;

creating, by the virtual browser, a mapping from a token of the first tree to the subtree; responsive to a determination that the cached version of the HTML page does not exist:

parsing, by the parsing engine of the virtual browser, the received HTML page to form a second syntax tree comprising nodes and tokens representing the non-matching content of the received data file, the second tree containing at least one static node; and

storing the second tree and the received HTML page in the cache.

9. (Cancelled)

10. (Cancelled)

11. (Currently Amended) A method for efficiently parsing received data files, comprising:

receiving, by a virtual browser executing on a device intermediary to a plurality of clients and a plurality of web servers, a first data file from a web server responsive to a request by a client of the plurality of clients;

retrieving a stored syntax tree from a cache, the stored syntax tree comprising nodes and tokens, representing data within the first data file and containing at least one static node and at least one token;

retrieving, by the virtual browser, a second data file from the cache, the second data file associated with the first data file;

identifying, by a comparison engine in communication with the virtual browser, non-matching content present only in the first data file;

parsing, by a parsing engine of the virtual browser, only the non-matching content of the first data file to form at least one subtree comprising nodes and tokens representing the non-matching content of the received data file; and

mapping, by the virtual browser, at least one of the tokens to at least one of the subtrees.

12. (Previously Presented) The method of claim 11, further comprising:

responsive to identifying non-matching content present only in the first file:
adding, by the virtual browser, at least one new token to the syntax tree.

13. (Currently Amended) A system for efficiently parsing input data from a plurality of content servers, comprising:

a virtual browser, deployed on a device intermediary to a plurality of clients and a plurality of web servers, for retrieving content from content servers;

an identification engine, in communication with the virtual browser for identifying retrieved content;

a cache, in communication with the virtual browser, for storing retrieved content and syntax trees comprising nodes and tokens representing data within the retrieved content;

a comparison engine in communication with the virtual browser, for comparing retrieved content with stored content to identify non-matching content not stored in the cache; and

a parsing engine of the virtual browser for parsing only the non-matching content identified by the comparison engine, forming subtrees comprising nodes and tokens representing the non-matching content of the received data file and creating a mapping from new tokens to formed subtrees.

14. (Canceled)

15. (Currently Amended) An intermediary for efficiently parsing received data files transmitted between a client and a server, the intermediary comprising:

a cache storing a version of a data file received from a server and a syntax tree comprising nodes and tokens representing data within the data file, the tree including at least one static node;

a comparison engine comparing the stored version of the data file with the received data file to identify non-matching content in the received data file; and

a virtual browser, executing on a device intermediary deployed between a plurality of clients and a plurality of web servers, in communication with the comparison engine, retrieving the stored version of the data file and the syntax tree from the cache, parsing only the non-matching content of the received data file to form at least one subtree comprising nodes and

tokens representing the non-matching content of the received data file, replacing at least one static node of the syntax tree with a token, and creating a mapping from each token to one of the subtrees.

16. (Currently Amended) A method for efficiently parsing received data files, the method comprising:

determining, by a virtual browser of a plurality of virtual browsers of a service executing on a computing device intermediary to a plurality of clients and a plurality of web servers, that a received web page comprises an object not stored in a cache;

identifying, by the virtual browser of the service responsive to a rule, that the object is to be tracked;

parsing, by the virtual browser of the service responsive to identifying that the object is to be tracked, content of the received web page to create an abstract syntax tree;

storing, by the virtual browser of the service, the abstract syntax tree and the content to the cache;

determining, by the virtual browser of the service, that the object of a second received web page is stored in the cache;

retrieving, by the virtual browser of the service responsive to determining that the object is stored in the cache, the abstract syntax tree and the content from the cache;

comparing, by the virtual browser of the service, the second received web page to the content to identify non-matching content in the second web page;

parsing, by the virtual browser of the service, only the non-matching content to generate a subtree; ~~and~~

modifying, by the virtual browser of the service, the abstract syntax tree to comprise a token mapped to the subtree; ~~and~~.

transmitting, by the service, the modified abstract syntax tree to a client responsive to a request by the client for the web page.

17. (Previously Presented) The method of claim 16, wherein parsing content of the received web page to create an abstract syntax tree further comprises designating each node in the abstract syntax tree as a static node.

18. (Previously Presented) The method of claim 16, further comprising identifying by the token dynamic content in the abstract syntax tree.